DOCUMENT RESUME

ED 424 265 TM 029 138

AUTHOR van der Linden, Wim J.
TITLE Optimal Assembly of Educational and Psychological Tests, with a Bibliography. Research Report 98-05.
INSTITUTION Twente Univ., Enschede (Netherlands). Faculty of Educational Science and Technology.
PUB DATE 1998-00-00
NOTE 38p.
AVAILABLE FROM Faculty of Educational Science and Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.
PUB TYPE Information Analyses (070) -- Reference Materials - Bibliographies (131)
EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Algorithms; Computer Assisted Testing; *Educational Testing; Equated Scores; Foreign Countries; Heuristics; *Item Banks; Item Response Theory; Linear Programming; *Psychological Testing; *Test Construction; Test Items

ABSTRACT

The advent of computers in educational and psychological measurement has lead to the need for algorithms for optimal assembly of tests from item banks. This paper reviews the literature on optimal test assembly and introduces the contributions to this report on the topic. Four different approaches to computerized test assembly are discussed: heuristic-based test assembly; 0-1 linear programming; network-flow programming; and an optimal design approach. In addition, applications of these methods to a large variety of problems are examined, including: (1) item response theory-based test assembly; (2) classical test assembly; (3) assembling multiple test forms; (4) item matching; (5) observed-score equating; (6) constrained adaptive testing; (7) assembling tests with item sets; (8) item pool design; and (9) assembling tests with multiple traits. This paper concludes with a 90-item bibliography on test assembly. (Contains three figures and seven references.) (Author/SLD)

**********
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
**********

TM

# Optimal Assembly of Educational and Psychological Tests, with a Bibliography

**Research Report 98-05**

ED 424 265

Wim J. van der Linden

U.S. DEPARTMENT OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

J. Nelissen

1

TM029138

faculty of
EDUCATIONAL SCIENCE AND TECHNOLOGY

University of Twente

Department of
Educational Measurement and Data Analysis

2

# Optimal Assembly of Educational and Psychological Tests, with a Bibliography

**Wim J. van der Linden**

This paper has been written as the introductory article to a forthcoming Applied Psychological Measurement Special Issue on Optimal Test Assembly.

**Abstract**

The advent of computers in educational and psychological measurement has led to the need of algorithms for optimal assembly of tests from item banks. This paper reviews the literature on optimal test assembly and introduces the contributions to this special issue on the topic. Four different approaches to computerized test assembly are discussed: heuristic-based test assembly, 0-1 linear programming, network-flow programming, and an optimal design approach. In addition, applications of these methods to a large variety of problems are examined, including IRT-based test assembly, classical test assembly, assembling multiple test forms, item matching, observed-score equating, constrained adaptive testing, assembling test with item sets, item pool design, and assembling tests with multiple traits. The paper concludes with a bibliography on optimal test assembly.

**Optimal Assembly of Educational and Psychological Tests, with a Bibliography**

In his chapters on item response theory (IRT) in Lord and Novick (1968), Birnbaum introduced a method of test assembly that was immediately acclaimed to be the proper approach to the problem. The method involves the following three steps: First, a goal for the test is formulated. Examples of possible goals are: admission decisions to an educational program, diagnosis of the skills of the students in the lower tail of a population distribution, or replacement of a test that has become obsolete by a parallel form. Second, the goal for the test is used to set a target for the test information function. Examples of such targets are given in Figure 1. Third, a test is assembled such that its information function matches the target. In

[Insert Figure 1 about here]

doing so,the fact is used that the item information functions are additive. Formal definitions of the concepts of item and test information are given later in this paper.

In spite of its immediate recognition, it took a long time before Birnbaum's method was actually used in the practice of test assembly. One reason for this delay was the fact that the method could not be performed by hand. But even when computers became available, it appeared difficult to formulate algorithms guaranteeing the optimality of a test assembled from an item pool. Finally, and most importantly, in practice tests are seldom assembled only to match a target for their information function but also have to meet large sets of specifications dealing with such attributes as test content, item format, cognitive level, or section lengths. In the early days of computerized testing it was not known how to implement Birnbaum's method to meet such specifications as well.

However, the formal structure of the above test assembly problem is not unique and can be found in many problems in industry, trade, commerce, and everyday life. Examples are the problems of putting together an investment portfolio, composing a diet, drafting a production schedule, packing a suitcase, or purchasing goods in a supermarket. The structure shared by these problems is the one of constrained combinatorial optimization (Nemhauser & Wolsey, 1988; Rao, 1985; Wagner, 1972). Each problem belonging to this class is characterized by the presence of a finite pool of "items" (e.g., stocks, nutrients, travel attributes) from which a combination has to be selected (e.g., portfolio, diet, contents of suitcase). The task is to select a combination of items that is optimal with respect to one attribute (e.g., maximum profit, maximum nutritional value, minimum weight) and at the same time meets a variety of constraints on other attributes of the problem (e.g., budget

available, minimum daily intake of vitamins and minerals, volume of suitcase). Problems of combinatorial optimization have been studied in decision theory, operations research, statistics, and management science.

To present test assembly as an example of constrained combinatorial optimization, an important distinction is made between the following two types of test specifications:

1. Constraints. These specifications require a test attribute or a function of item attributes to meet an upper and/or a lower limit. Constraints can be formulated as mathematical (in)equalities.
2. Objectives. These specifications require a test attribute or a function of item attributes to take a minimum or maximum value. Objectives can be formulated as mathematical functions that are to be optimized.

A test assembly program is now defined as a combination of an objective with a set of constraints. An example of a small IRT-based test assembly program is given in Figure 2.

[Figure 2 about here]

Observe that this program has three different types of constraints:

1. Constraints on categorical item attributes (e.g., content classification; use of graphics). These attributes partition the item pool, and the constraints hold for the distribution of the items over this partition.
2. Constraints on quantitative item attributes (word counts; expected response times). Constraints of this type require a function of the attributes (usually a sum or an average) over a set of items to meet an upper or lower bound.
3. Constraints on dependencies between items. Examples are constraints representing a relation of exclusion (mutually exclusive items) or inclusion between the items (e.g., items presented as sets with a common stimulus).

In practice, test assembly problems may involve many more attributes than the five attributes used in this example (for a catalogue, see van der Linden and Boekkooi-Timminga (1989).

Each possible objective involves its own optimal combination of items for a given item pool. Test assembly programs can therefore optimize only one objective function at a time. On the other hand, the number of constraints is not limited by any a priori bound. The only requirement is that the set of constraints leave a non-empty set of feasible solutions, that is, collections of items meeting each of the constraints. In principle, a large set of constraints can do so, but an inadvertently chosen small set can already overconstrain the problem and lead to infeasibility. Problems of infeasibility in test assembly models are analyzed in

Timminga and Adema (1996) and in the contribution by Timminga (1998) to this special issue.

Often, the same test assembly problem can be formulated as a variety of programs. For example, an important decision is whether or not to formulate a specification as an objective or a constraint. If, for a given item pool, the maximum value of the test information function at $\theta_0$ is approximately known, the objective function in Figure 2 can be replaced by a constraint that requires information at this point to be larger than a well-chosen lower bound. This replacement would allow another constraint to be formulated as the objective. Also, it is possible to join several test specifications into a weighted combination of functions of different item attributes and use this combination as an objective. Other choices emerge if a test assembly program is translated into a mathematical optimization model; examples of such choices will be met later in this paper.

## Basic Approaches

To find a solution to a test assembly program, a computer algorithm is needed. Four different approaches to solving test assembly programs will be discussed. Each of these approaches is represented by one or two contributions to this special issue of the journal. The first approach is based on the use of an intuitively attractive heuristic. This approach does not involve any mathematical modeling of the assembly program but formulates an item-selection rule that is built into a computer program. In the second and third approach, decision variables for the selection of the items for the test are defined. The variables are used to model the assembly problem as a mathematical programming problem with an objective function and constraints. An algorithm is then used to solve the model for an optimal combination of values for the decision variables. The fourth approach is based on the optimal design approach in statistics. This approach does not involve any combinatorial optimization but calculates a distribution of parameter values over a theoretic range that would yield a test with an optimal value for an objective function. These four approaches, combinations of which are often used in practice, are now discussed in more detail.

### Heuristic-Based Test Assembly

Most heuristics in the literature on test assembly are based on sequential item selection. That is, they select one item at a time, and the selection process is stopped when a

sufficient number is reached. These heuristics also belong to a class known as greedy heuristics in the optimization literature (e.g., Nemhauser & Wolsey, 1985, sect. II.5). The only other class of heuristics that has received some interest in the test assembly literature are those based on genetic algorithms (Michalewicz, 1994).

The basic nature of the greedy heuristic can be illustrated using the exemplary test assembly program in Figure 2. Indices i=1,...,I and j=1,...,n are used to denote the items in the pool and in the test to be assembled, respectively. Thus, $i_j$ is the index in the pool of the jth item in the test. Suppose j-1 items have been selected; the indices of these items form the set $S_{j-1} \equiv \{i_1,...i_{j-1}\}$. Therefore, $R_j \equiv (1,...,I)\backslash S_{j-1}$ is the set of items in the pool from which the jth item has to be selected. Finally, let $I_i(\theta)$ denote Fisher's information in item i on the unknown parameter $\theta$ (for a formal definition of this measure, see Lord, 1980, chap. 5).

If the test has to have maximum information at $\theta_0$, a greedy heuristic would select each next item to have maximum information at this value. It would be based on the following criterion:

$$i_j \equiv \max_t\{I_t(\theta_0): t \in R_j\}. \qquad (1)$$

To meet the categorical constraints in Figure 2, sets $R_j$ could be defined for each of the classes of the partition defined by the attributes. Item selection could then cycle along these classes proportionally to the numbers needed from them. Constraints on quantitative attributes or dependencies between items are more difficult to deal with in heuristics. The contributions by Luecht (1998) and Sanders and Verschoor (1998) to this special issue are based on the use of a greedy heuristic.

One of the first heuristics for IRT-based test assembly in the literature is given in Ackerman (1989; see also Wang & Ackerman, 1998). The heuristic has been designed to assemble a set of parallel test forms to meet a common target for their information functions but will be discussed here for the case of assembling a single form. It is assumed that test information is controlled at a series of discrete values $\theta_k$, k=1,...,K, where $T(\theta_k)$ is the target value for the test information function at $\theta_k$. At each step, the heuristic first selects the value of k for which the difference between current information and its target value is maximal. Then the item with maximum information at this value is selected. Let $k_j$ denote the index of the value of

$\theta$ used to select item j. Then, for j=1,...,n, the item selection process cycles through the following two criteria:

$$k_j \equiv \max_s\{T(\theta_s) - \sum_{i \in S_{j-1}} I_i(\theta_s);\ s=1,...,K\}, \tag{2}$$

$$i_j \equiv \max_t\{I_t(\theta_{k_j});\ t \in R_j\}. \tag{3}$$

A problem with Ackerman's heuristic is that the test information function is likely to overshoot its target for several $\theta$ values--a result typical of greedy heuristics. Luecht and Hirsch (1992) present a heuristic of a more tempered nature. Like (2), their heuristic is based on the difference between current information at $\theta_k$ and its target value. However, it divides the difference by the remaining number of items to be selected, n-j+1:

$$\delta_j(\theta_k) \equiv [T(\theta_k) - \sum_{i \in S_{j-1}} I_i(\theta_k)] / (n-j+1) \tag{4}$$

The quantities $\delta_j(\theta_k)$ are used as target values for the information function in the selection of the jth item:

$$i_j \equiv \min_t\{\sum_{k=1}^{K} w_j(\theta_k) | I_t(\theta_k) - \delta_j(\theta_k)|;\ t \in R_j\}, \tag{5}$$

where the weights $w_j(\theta_k)$ in (5) are added to promote the selection of items contributing most at $\theta$ values with large gaps between item information values and the targets. A more detailed introduction to this heuristic and the way it deals with various types of constraints on item selection is given in the contribution by Luecht (1998) to this special issue.

The heuristic by Swanson and Stocking (1993) supposes that <u>all</u> test specifications have been formulated as constraints. The heuristic minimizes a weighted sum of expected deviations from the constraints. Constraint 5 in Figure 2 is taken as an example, where $w_i$ is used to denote the number of words in item i. If the jth item is selected and item $t \in R_j$ is the candidate, the expected number of words in the total test is defined as:

$$\sum_{i \in S_{j-1}} w_i + w_t + \frac{n-j}{I-j} \sum_{i \in R_j \setminus \{t\}} w_i \qquad (6)$$

The first term in (6) is equal to the number of words in the j-1 items already selected, the second term is the number of words in candidate item t, and the last term is equal to n-j times the average number of words in the set of remaining items in set $R_j$. The expression in (6) is thus derived under the assumption of choosing item t and random sampling of the rest of the items from set $R_j \setminus \{t\}$.

The Swanson-Stocking heuristic calculates these expected values for all constraints. It then calculates the extent to which these expectations violate the bounds in the constraints. Finally, a weighted sum of the deviations is calculated, and the item with the smallest value for the weighted sum is selected. The use of weights not only allows us to express preferences for constraints but is also necessary to compensate for scale differences between attributes and bounds.

As already noted, the only addition to the class of greedy heuristics for test assembly are those based on genetic algorithms (Verschoor, 1998). Genetic algorithms do not select items sequentially. They start with a pool of candidate solutions for the full test that are improved in a probabilistic way simulating an evolutionary process. A key feature of genetic algorithms is that they have a nonzero probability of backtracking. Greedy heuristics, on the other contrary, make choices that are locally optimal but may end up with solutions that are not globally optimal. These heuristics are therefore often followed by a second process in which some of the items in the solution are replaced by alternatives. For example, Ackerman (1989) recommends swapping items between multiple forms to improve the extent to which they are parallel. Likewise, Swanson and Stocking (1993) recommend a second stage in which items whose removal would result in a reduction of the weighted sum of deviations are replaced by more promising ones.

0-1 Linear Programming

As already noted, the critical difference between this approach and the previous one is the definition of decision variables to assign items from the pool to the test. These variables are used to model the objective as a mathematical function and the constraints as (in)equalities to be imposed on its optimization. An example is formulated for the test assembly program in

Figure 2.

Let $x_i$, i=1,...,I, be the variable to represent the decision whether ($x_i$=1) or not ($x_i$=0) to assign item i from the pool to the test. The sets of indices of the items in the pool on knowledge of fact, applications, and with graphics will be denoted as $V_k$, $V_a$, and $V_g$, respectively. In addition to the quantitative attribute $w_i$ for the number of words in item i, the attribute $r_i$ is used for the expected response time on item i.

The model is as follows:

$$\text{maximize} \sum_{i=1}^{I} I_i(\theta_0)\, x_i \qquad \text{(maximum information at } \theta_0) \qquad (7)$$

subject to

$$\sum_{i \in V_k} x_i \le 10, \qquad \text{(knowledge of facts)} \qquad (8)$$

$$\sum_{i \in V_a} x_i \ge 10, \qquad \text{(applications)} \qquad (9)$$

$$\sum_{i \in V_g} x_i = 5, \qquad \text{(graphics)} \qquad (10)$$

$$\sum_{i=1}^{I} x_i = 25, \qquad \text{(test length)} \qquad (11)$$

$$\sum_{i=1} w_i\, x_i \le 1{,}500, \qquad \text{(word counts)} \qquad (12)$$

$$\sum_{i=1}^{I} r_i\, x_i \le 60, \qquad \text{(expected response times)} \qquad (13)$$

$$x_{64}+x_{65} \le 1, \qquad \text{(mutually exclusive items)} \qquad (14)$$

$$x_i \in 0,1, \quad I = 1,\ldots,I \qquad \text{(range of variables)} \qquad \textbf{(15)}$$

Since the variables are zero-one, the sum in (7) is the information in the test at $\theta_0$. Likewise, the sums in (12) and (13) are the total number of words and the expected response time for the test, respectively. In (8)-(10), the sums of variables are the numbers of items in the test form the various sets; in (15) this sum represents the length of the test.

The expressions in (7)-(14) are linear in the variables. The constraints in (11) are technical constraints that define the range of the variables. The optimization problem therefore belongs to 0-1 linear programming (LP). Optimal values for the decision variables $x_i$, i=1,...,I, can be found using standard LP software or a dedicated test assembly software package such as ConTEST (Timminga, van der Linden & Schweizer, 1996). Exact solutions to 0-1 LP problems are obtained through a complete branch-and-bound (B&B) search. Such searches are known to be NP-hard; that is, their solution time is not bounded by a polynomial of the size of the problem. Exact solutions of large problems may therefore require an excessive amount of time. However, solutions with values for the objective function differing from the optimum by a predetermined, negligibly small factor can easily be obtained for item pools of a realistic size. An algorithm for doing so is the described in by Adema, Boekkooi-Timminga and van der Linden (1991; see also Timminga, van der Linden & Schweizer, 1996, sect. 6.6.5). The algorithm fixes some of the decision variables using a result in Crowder, Johnson and Padberg (1983). In addition, the value of the objective function in the solution to the relaxed problem, that is, with the 0-1 variables replaced by variables that can take values in [0,1], is employed to derive a stopping rule for a B&B search for the solution in the original problem. Fan (1997) used the algorithm to assemble six parallel forms of 60 items, each with approximately 200 constraints, from a pool of nearly 3,000 items within 11 mins.

To the knowledge of the author, the first to apply linear programming to model a problem in testing was Votaw (1952). Feuermann and Weiss (1973) used the technique to solve a test assembly program. The application of LP linear programming to test assembly was also alluded to in Yen (1983). A seminal paper was the one by Theunissen (1985) who modeled Birnbaum's problem of a test to meet a target information function as a 0-1 LP problem. This paper stimulated others to use the same methodology to model a large variety of other test assembly problems (see the papers by Adema, Baker, Boekkooi-Timminga, Boomsma, de Gruijter, Gademann, Glas, Kester, Razoux Schultz, Timminga, and van der

Linden in the bibliography at the end of this paper). In this special issue, the paper by van der Linden and Reese (1998) demonstrates the use of 0-1 LP to build item selection constraints into an algorithm for computerized adaptive testing.

Network-Flow Programming

Integer programming problems are defined as LP problems with decision variables that can take a larger range of integer values than just the values of 0 and 1. In special cases, integer problems take the form of a network-flow or transportation problem. If so, quick solutions to large problems are possible. An example of a problem with a network-flow structure is given by the directed graph in Figure 3. Nodes $S_i$ on the left-hand side are supply

[Figure 3 about here]

nodes; nodes $D_j$ on the right-hand side demand nodes. The directed arcs or arrows indicate a flow or transportation from the supply to the demand nodes. For each arc there is a decision variable $x_{ij}$ denoting the units of flow from node $S_i$ to $D_j$. The constraints in a network-flow problem deal with the numbers of units available at the supply nodes, the bounds on the numbers needed at the demand nodes, or the costs associated with a units of flow along the arc from i to j, $c_{ij}$. If the number of supply nodes is equal to the number of demand nodes and the decision variables take only the values 0 and 1, network-flow problems are known as assignment problems. Also, transhipment nodes can be added between the supply and demand nodes to accommodate a larger class of problems. Transshipment nodes have both demand and supply constraints associated with them.

An important result in network-flow programming is that among the solutions to the relaxed or continuous version of the problem there is always one with integer values for the variables. This solution is found by the well-known simplex algorithm in LP. Moreover, the structure of the network-flow problems allows for an efficient implementation of the simplex algorithm resulting in solution times for large problems that seldom take more than seconds on a personal computer.

Some test assembly problems can be formulated as network-flow problems. For example, suppose that for $i=1,\ldots,n_1$ supply nodes $S_i$ represent the items in the example in Figure 2 that measure knowledge of facts whereas for $i=n_1+1,\ldots,I$, they represent the items that do not measure at this cognitive level. In addition, demand nodes $D_j$, $j=1,2$, represent the sets of items needed in the test form that do and do not measure knowledge of facts, respectively. The decision variables $x_{ij}$ denote whether ($x_{ij}=1$) or not ($x_{ij}=0$) item i is shipped to the part of

the test represented by demand node $D_j$. Finally, the "cost" of shipping item i to $D_j$ is defined as its information at $\theta_0$, $I_i(\theta_k)$ (changing the problem from a minimization into a maximization problem). The test assembly problem consisting of the objective and the first constraint in Figure 2 can be modeled as the following network-flow problem:

$$\text{maximize} \sum_{i=1}^{I} I_i(\theta_0)x_{ij} \qquad \text{(maximum information at } \theta_0\text{)} \qquad \textbf{(16)}$$

subject to

$$\sum_{j=1}^{2} x_{ij} \leq 1, \quad i=1,\ldots,I, \qquad \text{(supply at } S_1,\ldots,S_I\text{)} \qquad \textbf{(17)}$$

$$\sum_{i=1}^{n1} x_{i1} = 10 \qquad \text{(demand at } D_1\text{)} \qquad \textbf{(18)}$$

$$\sum_{i=n+1}^{I} x_{i2} = 15 \qquad \text{(demand at } D_2\text{)} \qquad \textbf{(19)}$$

$$x_{ij} \in \{0,1\}, \; i=1,\ldots,I, \; j=1,2, \qquad \text{(range of variables)} \qquad \textbf{(20)}$$

where $x_{i1}=0$ for $i>n_1$ and $x_{i2}=0$ for $i\leq n_1$.

Most test assembly problems with categorical attributes can be modeled as network-flow problems with demand nodes representing classes of items defined by combinations of attributes. Since these classes need not form a partition of the item bank and transshipment nodes can be added, flexibility is large. The fact that realistic problems typically may involve thousands of variables (number of items times number of demand nodes) need not bother us; such network-flow problems can generally be solved quickly.

However, problems with quantitative attributes are more difficult to model. One approach is to embed the network-flow problem in a heuristic, for example, using Lagrangian relaxation. In this technique, all quantitative constraints are removed from the constraint set and added to the objective function as penalty terms times a Lagrange multiplier. For

example, Constraint 5 in Figure 2 is added to the objective function in (16) as:

$$\text{maximize} \sum_{i=1}^{I} I_i(\theta_0)\, x_{ij} - \lambda(1{,}500 - \sum_{i=1}^{I} w_i\, x_i). \qquad (21)$$

A solution is typically found cycling through the process of finding a suitable value for $\lambda$, solving the network-flow problem, and improving on the current value of $\lambda$ until a satisfactory result is obtained. Results are usually still quick and near optimal but may suffer from constraint violation.

Test assembly problems with constraints representing dependencies between items in the pool can not always be formulated as network-flow problems either. However, the same approach of embedding a reduced problem in a larger heuristic can be followed to attack such problems.

An excellent review of network-flow programming models with Lagrangian relaxation for test assembly is given in Armstrong, Jones and Wang (1995). Nearly all of their empirical examples have calculation times less than 2 mins. In the contribution by Armstrong, Jones and Kunce (1998) to this special issue, the same technique is used to assemble a series of parallel test forms. Other applications are given in the papers by Armstrong et al., Boomsma, and Veldkamp in the bibliography.

Optimal Design Approach

The final approach reviewed here is based on the theory of optimal experimental design developed in statistics (e.g., Fedorov, 1972). One of the first problems addressed in optimal design theory was the designing of an experiment for estimating the parameters in a linear regression model. The standard approach in optimal design theory is to choose a set of design points (=grid of values for the independent variables) and find an experimental design (=distribution of observations over these points) that would result in optimal accuracy of the parameter estimates. Since most experiments have multiple parameters, the criterion of optimality is typically defined on the variance-covariance matrix of the estimators. Popular functions are the determinant, the trace, and the eigenvalue of the this matrix; solutions with optimal values for these criteria are known as D-, A-, and L-optimal, respectively.

Since IRT models can be viewed as regression models, it seems obvious to apply the techniques of optimal design to parameter estimation problems in IRT. Applications consists

of optimal design of experiments for estimating the item as well as the examinee parameters. The latter is the problem of optimal test design. A solution to the problem is a joint distribution of the item parameter values with optimal accuracy for the ability estimator. However, unlike standard regression models, IRT models are nonlinear and have unobserved independent values. How to deal with these issues is explained in the reviews of optimal design approaches to IRT by Berger (1997) and van der Linden (1994b). The contribution by Berger (1998) to this special issue of the journal applies optimal design techniques to tests with dichotomous and polytomous item formats. Other applications of optimal test design are given in the papers by Berger et al. in the bibliography.

Discussion

Important yardsticks to evaluate the appropriateness of the various approaches to test assembly problems are: (1) easiness of modeling the problem; (2) optimality of the solution; (3) possibility of constraint violation; and (4) computer time needed. A heuristic approach is generally quicker than all other approaches. However, its solutions are mostly suboptimal to an extent that remains unknown and may violate some of the constraints. As already observed, the use of heuristics does not involve any modeling but for new problems it usually takes a considerable amount of time to adjust the heuristic, for example, to find best weights if the objective is to minimize a sum of weighted deviations form a large set of constraints.

The strong advantage of the 0-1 LP approach is its flexibility. Most assembly problems can be modeled using 0-1 integer variables. Also, modeling is the only thing needed; once a model has been formulated, it is not necessary to design a heuristic or adjust software. Constraint violation is impossible. However, the approach does have an important tradeoff between the speed and optimality of its solutions. Exact solutions for larger problems are not possible in realistic time, but if an appropriate search algorithm is used, near-optimal solutions to practical problems, with values for the objective function 1-2% from its optimum, say, are often possible in minutes.

The power of network-flow programming is its speed. If the test assembly problem can be formulated to have the special structure of a network-flow problem, exact solutions to large problems are possible in seconds. If not, the method has to be embedded in a heuristic approach. Typically, solutions then still take seldom more than a few minutes but are near optimal and may show occasional constraint violation.

The optimal design approach differs from the others in several aspects. Its intention is

to calculate the best distribution of the item parameters values over their theoretical range given a criterion of optimality. Other test specifications than this objective are generally ignored. Optimal design is thus not a method for assembling a test from an finite, existing pool of items. However, its optimal distribution of item parameter values should be approximated in practice. In principle, it is even possible to build this distribution as a target in a 0-1 LP or network-flow model for test assembly.

## Applications

A large variety of test assembly problems have been attacked using the approaches discussed in this paper. Applications range from the problem of assembling a set of multiple test forms simultaneously to observed-score equating and constrained adaptive testing. The most important results are now reviewed.

Multiple forms. The first extension of the problem of finding an optimal single test form was the one of assembling a set of parallel forms. An obvious approach to the problem of multiple-form assembly may seem to apply the above approaches sequentially until the desired number of forms is obtained. However, this approach would select the best items first and show a decrease in the qualify of the test forms. Therefore, simultaneous assembly of multiple test forms is a better alternative.

As shown in Boekkooi-Timminga (1987a), a simultaneous approach involves replacing the decision variables in the model in (7)-(15) by variables $x_{if}$ denoting the decisions whether ($x_{if}=1$) or not ($x_{if}=0$) item i in the pool will be assigned to form f=1,...,F. In addition, a set of constraints has to be added to prevent items from being assigned to more than one form:

$$\sum_{f=1}^{F} x_{if} \leq 1, \quad i=1,...,I. \tag{22}$$

Because the number of decision variables is equal to the size of the item pool times the number of forms, the approach is only possible for smaller problems. All developments for realistic multiple-form problems therefore have heuristic aspects.

Adema (1992b) designed an approach in which the problem of assembling a set of parallel forms simultaneously is replaced by a series of computationally less intensive two-

form problems. A generalization of the approach to any set of test forms is given in van der Linden and Adema (1998). Other methods of assembling multiple test forms are proposed in Adema (1992), Boekkooi-Timminga (1990a; 1990b) and van der Linden and Carlson (1997). Solutions based on item matching are given Armstrong, Jones, Li and Wu (1996), Armstrong, Jones and Wu (1992) and in the contribution by Armstrong, Jones and Kunce (1998) to this special issue. The principle of item matching used in the Armstrong et al. papers is explained below.

Item sets. A popular testing format is the one with sets of items related to a common stimulus, for example, a text passage in a reading test or a description of an experiment in a physics test. If each item set in the pool remains intact if selected for a test, an obvious approach is to attach aggregated item attributes as descriptors to the item sets and model the problem using 0-1 decision variables for the selection of sets. The problem becomes more complicated though if the number of items to be selected per set has to be smaller than the number in the pool, in particular if the selection also has to satisfy separate sets of constraints on item, test, and stimulus attributes.

A flexible solution is possible using different decision variables for the stimuli and the items (van der Linden, 1992). Let $s=1,...,S$ denote the stimuli in the pool and $i_s=i=1,...,I_s$, the items nested under stimulus s. These indices can be used to define 0-1 decision variables $z_s$ and $x_{i_s}$ for the selection of the stimuli and items, respectively. The same variables are then available to model the various specifications at item, test, and stimulus level. They also allow for the simultaneous selection of stimuli and items provided the following constraint set is added to the model

$$\sum_{i_s=1}^{I_s} x_{i_s} - n_s z_s = 0, \quad s=1,...,S. \tag{23}$$

The purpose of these constraints, which can be replaced by inequalities, is not only to keep the selection of stimuli and items consistent but also to set the number of items selected per set equal to $n_s$.

Classical test assembly. A basic problem in test assembly based on classical item and test parameters is that, unlike IRT, no meaningful test parameters can be found that are additive in the items. In particular, test reliability is a nonlinear function of the covariances

between all pairs of items, and, as a consequence, an attempt to assembly a test with maximum reliability may involve a procedure with endless backtracking.

The problem of nonlinearity is illustrated for the maximization of Cronbach's alpha. Adding decision variables, the objective function is

$$\text{maximize } \alpha = \frac{n}{n-1}\left[1 - \frac{\sum_{i=1}^{I} \sigma_i^2 x_i}{\sum_{i=1}^{I} \rho_i \sigma_i x_i}\right], \tag{24}$$

where $\sigma_i$ and $\rho_i$ are the item standard deviation and item-test correlation, respectively. However, if the test length, n, is fixed, the objective is equivalent to the one of minimizing the ratio in the second factor. Also, both the numerator and denominator of this ratio are linear in the decision variables. Adema and van der Linden (1989) presented an LP solution in which the numerator is maximized and the denominator is constrained to be lower than a well-chosen small bound, c:

$$\text{maximize } \sum_{i=1}^{I} \rho_i \sigma_i x_i \tag{25}$$

subject to

$$\sum_{i=1}^{I} \sigma_i^2 x_i \leq c. \tag{26}$$

Simulation studies with this linearized version of Cronbach's alpha showed near-optimal results under a large variety of conditions. Armstrong, Jones and Wang (1994) extended the approach by building the constraint in (26) into the objective function in (25) using Lagrangian relaxation and embedding the new objective function into an algorithm that optimized the choice of c.

Item matching. Problems of item matching arise if a set of test forms has to be assembled that are indistinguishable item by item. The first application of optimal test assembly methods to such problems was the use of 0-1 LP to find optimally matched test

halves for estimating split-half reliability (van der Linden and Boekkooi-Timminga, 1988). The same problem has been addressed using network-flow programming in Armstrong and Jones (1992) and in the contribution by Sanders and Verschoor (1998) to this special issue who use a greedy heuristic.

A related problem is the one of assembling a set of test forms to be parallel to an old form addressed in Armstrong, Jones, Li and Wu (1996), Armstrong, Jones and Wu (1992) and in the contribution by Armstrong, Jones and Kunce (1998) to this special issue. Network-flow programming is a natural approach to this problem because the items in the reference test can serve as demand nodes to which items for the set of forms are shipped at costs that are a function of the match between the items and the target (see Figure 3). Once the items have been shipped, a heuristic is used to assign the items from the demand nodes to the individual test forms.

Observed-score equating. In large-scale testing programs old test forms are periodically replaced by new ones. The traditional approach is to assemble a new form, pretest its items, and equate the observed scores on the new form to those on the old form. An alternative would be to assemble the new form to have the same observed-score distribution as the old form for a population of examinees. The idea was explored in van der Linden and Luecht (1996) using an 0-1 LP model that matched both the test information and the test characteristic function of the new form to those of the old form, the idea being that these two functions would equate the error- and true-score distributions of the new form, and thereby its observed-score distribution. The same idea is used in Glas (1988) to equate cutscores on a new and old form and in the contribution by Armstrong, Jones and Kunce (1998) to this special issue of the journal.

In a later paper (van der Linden & Luecht, in press), it is proved that the observed-score distributions on two test forms are equal if and only if

$$\sum_{i=1}^{n} P_i^r(\theta) = \sum_{j=1}^{n} P_j^r(\theta), \quad \text{for } r=1,\ldots,n, \tag{27}$$

where $P_i(\theta)$ and $P_j(\theta)$ are the response functions of item i and j in the new and old form, respectively. The result is based on a series expansion and in practice only a few lower-order equalities need to be met to get good results. Since the equalities are linear in the items, they can easily be built in a 0-1 LP model for assembling the new form. An empirical example for the

LSAT gave excellent results for a model that only had the equalities in (27) for r=1,2,3.

Constrained adaptive testing. Though the development of computerized adaptive testing was motivated by the idea of maximizing the statistical precision of ability estimation, real-life applications have shown the need of such tests to keep the content specifications constant across examinees as well. A 0-1 LP approach to adaptive testing in which the information in the test is maximized at the current ability estimate subject to a large set of constraints is presented in the contribution by van der Linden and Reese (1998) to the special issue of this journal. The algorithm starts with the on-line assembly of a full test that meets each of the constraints and is optimal at the initial ability estimate. Each next step, the most informative item from the test is administered and both the ability estimate and set of constraints are updated. An example for the LSAT shows that several hundred constraints can be built into the item selection procedure without sacrificing any precision of the ability estimator. A comparable approach based on network-flow programming was developed independently in Cordova (1997). An application of the algorithm with response-time constraints used to control adaptive tests for differential speededness between examinees is presented in van der Linden, Scrams and Schnipke (submitted).

Assembling multidimensional tests. For larger item pools, a potential problem with the use of the simple logistic IRT models for item calibration is violation of their assumption of unidimensionality. If so, a multidimensional IRT model has to be used. However, for a model with multiple ability parameters test information is not a scalar, and the variance-covariance matrix of the estimators has to be addressed directly. Test assembly can then no longer follow Birnbaum's method based on a target for the test information function.

A 0-1 LP-based algorithm for multidimensional test assembly is given in van der Linden (1996). The model is based on a target for the variance functions of the ability estimators using the fact that, though not linear in the items themselves, these functions are built up of linear expressions. In the model, some of these expressions are optimized, others constrained. Repeated application of the model systematically varying the bounds in the constraints can be used to find a solution fitting the targets for the information functions best. An example for an item pool from the ACT Assessment Program yielded test forms meeting a uniform target for the variance functions over the ability space. A version of the approach with Lagrangian relaxation is given in Veldkamp (submitted).

Item pool design. The final application of optimal assembly methods in this review is the one to the problem of assembling an item pool. The importance of this application lies in

the fact that item pools in testing programs are not always on target. As a consequence, some portions of the item pool are quickly depleted whereas others may have items that are never used.

The problem of item pool design has been explored in Boekkooi-Timminga (1991). Her approach starts with a tentative blueprint for the item pool from which test forms are assembled to find out what types of items are over and underrepresented. The results are then used to adjust the blueprint. Another approach is followed in van der Linden, Veldkamp and Veldkamp (in preparation). The decision variables in their integer programming model represent the numbers of items in the pool needed and optimal values for the variables are found using an objective function that minimizes an empirical estimate of the costs involved in item writing.

## Concluding Remark

Modern measurement is characterized by the use of statistical models for the quantification of educational and psychological variables. As in any other quantitative field, an obvious next step is the application of optimization techniques to maximize the utility of the models. This special issue reviews a variety of applications of such techniques to the problem of optimal test assembly and presents several new applications. The mathematical techniques involved are neither new nor applied for the first time. However, what is new is the creativity involved in analyzing test assembly problems and structuring them such that the optimization techniques apply. Since most results are of recent date, it is anticipated that more will follow.

**References**

Crowder, H., Johnson, E.L., & Padberg, M. (1983). Solving large-scale linear programming problems: Some algorithmic techniques and computational results. Operations Research, 31, 803-834.

Fedorov, V.V. (1972). Theory of optimal experiments. New York: Academic Press.

Lord, F.M. (1980). Applications of items response theory to practical testing problems. Hillsdale, NJ: Erlbaum.

Michalewicz, Z. (1994). Genetic algorithms + data structures = evolution programs (2nd ed.). New York: Springer-Verlag

Nemhauser, G., & Wolsey, L. (1988). Integer and combinatorial optimization. New York: Wiley.

Rao, S.S. (1985). Optimization: Theory and applications. New Delhi: Wiley.

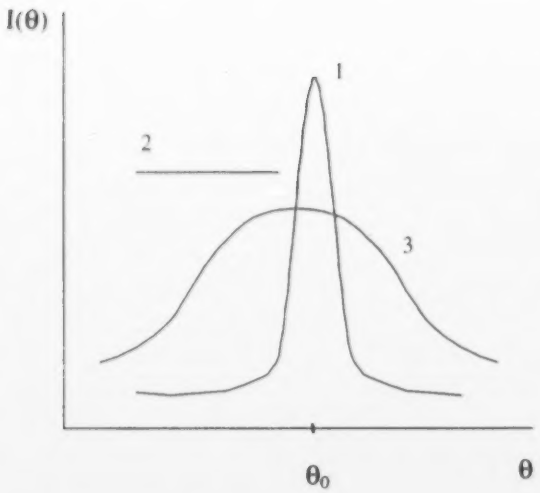
Wagner, H.M. (1972). Principles of operations research, with applications to managerial decisions. London: Prentice-Hall.

**Figure Captions**

Figure 1. Examples of targets for test information functions (1. Selection decision with cut score $\theta_0$; 2. Diagnostic test for low ability examinees; 3. Information function of an old test to be matched)

Figure 2. Example of a test assembly model or program.

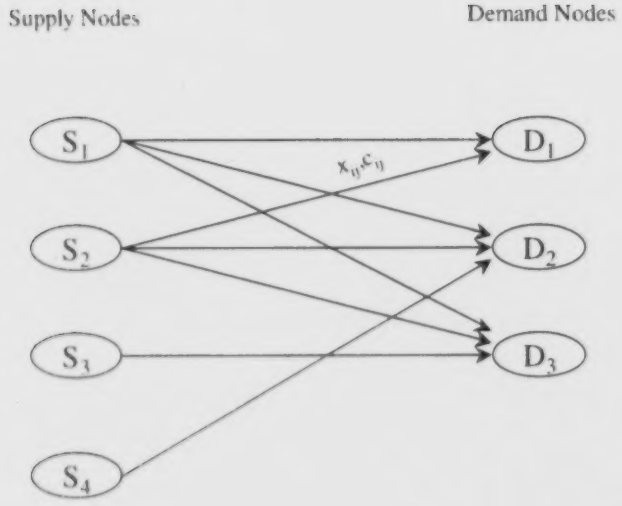
Figure 3. A directed graph of a network-flow programming problem


I(θ)
1
2
3
θ0
θ

Maximize test information at cut score

subject to

1. No more than 10 items on knowledge of facts;
2. At least 10 items on applications;
3. Five items with graphics;
4. Test length equal to 25 items;
5. Total number of words in test not larger than 1,500;
6. Total expected response time not larger than 60 minutes;
7. Items 64 and 64 not simultaneously in the test.


Supply Nodes
Demand Nodes
S1
S2
S3
S4
D1
D2
D3
x11,c11

**Bibliography on Optimal Test Assembly**[1]

Ackerman, T. (1989, March). An alternative methodology for creating parallel test forms using the IRT information function. Paper presented at the annual meeting of the National Council on Measurement in Education, San Francisco, CA.

Adema, J.J. (1988). A note on solving large-scale zero-one programming problems (Research Report No. 88-4). Enschede, The Netherlands: University of Twente, Department of Educational Measurement and Data Analysis.

Adema, J.J. (1990). The construction of customized two-staged tests. Journal of Educational Measurement, 27, 241-253.

Adema, J.J. (1990). Models and algorithms for the construction of achievement tests. Unpublished doctoral dissertation, University of Twente, Enschede, The Netherlands.

Adema, J.J. (1992). Methods and models for the construction of weakly parallel tests. Applied Psychological Measurement, 16, 53-63.

Adema, J.J. (1992). Implementations of the branch-and-bound method for test construction. Methodika, 6, 99-117.

Adema, J.J., Boekkooi-Timminga, E. & Gademan, A.J.R.M. (1992). Computerized test construction. In M. Wilson (Ed.), Objective measurement: Theory into practice. (Vol. 1, pp. 261-273). Norwood, New Jersey: Ablex.

Adema, J.J., Boekkooi-Timminga, E., & van der Linden, W.J. (1991). Achievement test construction using 0-1 linear programming. European Journal of Operations Research, 55, 103-111.

Adema, J.J. & van der Linden, W.J. (1989). Algorithms for computerized test construction using classical item parameters. Journal of Educational Statistics, 14, 279-290.

Armstrong, R.D. and Jones, D.H. (1992). Polynomial algorithms for item matching. Applied Psychological Measurement, 16, 365-373.

Amstrong, R.D., Jones, D.H., & Kunce, C.S. (1998). IRT test assembly using network-flow programming. Applied Psychological Measurement, 22. [This issue]

[1] This bibliography has been carefully composed. The author apologizes for any publication he might have missed.

Armstrong, R.D., Jones, D.H., Li, X., L., & Wu, I.-L. (1996). A study of network-flow algorithm and a noncorrecting algorithm for test assembly. Applied Psychological Measurement, 20, 89-98.

Armstrong, R.D., Jones, D.H., & Wang, Z. (1994). Automated parallel test construction using classical test theory. Journal of Educational Statistics, 19, 73-90.

Armstrong, R.D., Jones, D.H., & Wang, Z. (1995). Network optimization in constrained standardized test construction. In K.D. Lawrence Ed.). Applications of management science: Network optimization applications (Vol. 8, pp. 189-212). Greenwich, CT: JAI Press.

Armstrong, R.D., Jones, D.H., & Wu, I.-L. (1992). An automated test development of parallel tests. Psychometrika, 57, 271-288.

Baker, F.B., Cohen, A.S., & Barmish, B.R. (1988). Item characteristics of tests constructed by linear programming. Applied Psychological Measurement, 12, 189-200.

Berger, M.P.F. (1994). A general approach to algorithmic design of fixed-form tests, adaptive tests, and testlets. Applied Psychological Measurement, 18, 141-153.

Berger, M.P.F. (1997). Optimal designs for latent variable models: A review. In J. Rost & R. Langeheine (Eds.), Applications of latent trait and latent class models in the social sciences (pp. 71-79). Münster, Germany: Waxmann.

Berger, M.P.F. (1998). Optimal design of tests with items with dichotomous and polytomous response formats. Applied Psychological Measurement, 22. [This issue]

Berger, M.P.F., & Mathijssen, E. (1997). Optimal test designs for polytomously scored items. British Journal of Mathematical and Statistical Psychology, 50, 127-141.

Berger, M.P.F., & van der Linden, W.J. (1992). Optimality of sampling design in item response theory models. In M. Wilson (Ed.), Objective measurement: Theory into practice (pp. 274-288). Norwood, N.J.: Ablex Publishing Corporation.

Berger, M.P.F. & van der Linden, W.J. (1995). Het optimaal ontwerpen van tests met verschillende optimaliteitscriteria [Designing educational tests with different criteria of optimality]. Tijdschrift voor Onderwijsresearch, 20, 79-92.

Berger, M.P.F., & Veerkamp, W.J.J. (1996). A review of selection methods for optimal test design. In G. Engelhard, Jr. & M. Wilson (Ed.), Objective measurement: Theory into practice (Vol. 3, pp. 437-456). Norwood, New Jersey: Ablex Publishing Company.

Birnbaum, A. (1986). Some latent trait models and their use in inferring an examinee's ability. In F.M. Lord and M.R. Novick, Statistical theories of mental test scores. Reading, MA: Addison-Wesley.

Boekkooi-Timminga, E. (1987a). Simultaneous test construction by zero-one programming. Methodika, 1, 1101-112.

Boekkooi-Timminga, E. (1987b). Some methods for simultaneous test construction. In W.J. van der Linden (Ed.), IRT-based test construction (Research Report No. 87-2). Enschede, The Netherlands: University of Twente, Department of Educational Measurement and Data Analysis.

Boekkooi-Timminga, E. (1989). Models for computerized test construction. Unpublished doctoral dissertation, University of Twente, Enschede, The Netherlands.

Boekkooi-Timminga, E. (1990a). The construction of parallel tests from IRT-based item banks. Journal of Educational Statistics, 15, 129-145.

Boekkooi-Timminga, E. (1990b). A cluster-based method for test construction. Applied Psychological Measurement, 15, 129-145.

Boekkooi-Timminga, E. (1991, June). A method for designing Rasch model based item banks. Paper presented at the Annual Meeting of the Psychometric Society, Princeton, NJ.

Boekkooi-Timminga, E. (1993). Computer-assisted test construction. Social Science Computer Review, 11, 292-300.

Boekkooi-Timminga, E. & Sun, L. (1991). CONTEST: A computerized test construction system. In J. Hoogstraten & W.J. van der Linden (Eds.), Methodologie [Methodology] (pp. 69-76). Amsterdam, The Netherlands: SCO.

Boekkooi-Timminga, E., & van der Linden, W.J. (1988). Algorithms for automated test design. In F.J. Maarse, L.J.M. Mulder, W.P.B. Sjouw & A.E. Akkerman (Eds.), Computers in psychology: Methods, instrumentation and psychodiognostics (pp. 171-176). Berwyn, PA: Swets Publishing.

Boomsma, Y. (1986). Item selection by mathematical programming. Unpublished master's thesis, University of Twente, Enschede, The Netherlands.

Cordova, M.J. (1997). Optimization methods in computerized adaptive testing. Unpublished doctoral dissertation, Rutgers University, New Brunswick, NJ.

de Gruijter, D.N.M. (1990). Test construction by means of linear programming. Applied Psychological Measurement, 14, 175-181.

Fan, M. (1997, June). A comparison of computerized test assembly programs for constructing parallel test forms. Paper presented at the Annual Meeting of the Psychometric Society, Gatlinburg, TN.

Feuerman, F. & Weiss, H. (1973). A mathematical programming model for test construction and scoring. Management Science, 19, 961-966.

Gademann, A.J.R.M. (1987). Item selection using multi-objective programming (OIS Report No. 1). Arnhem, The Netherlands: Cito

Glas, C.A.W. (1988). Psychometric aspects of maintaining standards of examinations. Educational Psychology, 8, 257-270.

Kelderman, H. (1987). Some procedures to assess target information functions. In W.J. van der Linden (Ed.), IRT-based test construction (Research Report 87-2). Enschede, The Netherlands: University of Twente, Department of Educational Measurement and Data Analysis.

Kester, J.G. (1988). Various mathematical programming approaches toward item selection (OIS Project Report No. 3). Arnhem, The Netherlands: Cito.

Luecht, R.D. (1988). Computer-assisted test assembly using optimization heuristics. Applied Psychological Measurement, 22. [This issue]

Luecht, R.M. & Hirsch, T.M. (1992). Computerized test construction using average growth approximation of target information functions. Applied Psychological Measurement, 16, 41-52.

Miyaji, I., Nakagawa, Y, & Ohno, K. (1995). Decision support system for composition of the examination problem. European Journal of Operational Research, 80, 130-138.

Razoux Schultz, A.F. (1987). Item selection using heuristics (IOS Report No. 2). Arnhem, The Netherlands: Cito.

Sanders, P.F., Verschoor, A.J. (1998). Parallel test construction using classical item parameters. Applied Psychological Measurement, 22. [This issue]

Stocking, M.L. & Swanson, L. (1993). A method for severely constrained item selection in adaptive testing. Applied Psychological Measurement, 17, 277-292.

Stocking, M.L., & Swanson, L. (1998). Severely constrained adaptive testing with extensions to item pool design. Applied Psychological Measurement, 22. [This issue]

Stocking, M.L., Swanson, L., & Pearlman, M. (1993). Application of an automated item selection method to real data. Applied Psychological Measurement, 17, 167-176.

Swanson, L. & Stocking, M.L. (1993). A model and heuristic for solving very large item selection problems. Applied Psychological Measurement, 17, 151-166.

Theunissen, T.J.J.M. (1985). Binary programming and test design. Psychometrika, 50, 411-420.

Theunissen, T.J.J.M. (1986). Optimization algorithms in test design. Applied Psychological Measurement, 10, 381-389.

Theunissen, T.J.J.M. (1996). Combinatorial issues in test construction. Unpublished doctoral dissertation, University of Amsterdam, The Netherlands.

Theunissen, T.J.J.M., & Verstralen, H.H.F.M. (1986). Algoritmen voor het samenstellen van toetsen [Algorithms for test assembly]. In W.J. van der Linden (Ed.), Moderne methoden van toetsconstructie- en gebruik [Modern methods of test assembly and administration] (pp. 32-39). Amsterdam, The Netherlands: Swets & Zeitlinger.

Timminga, E. (1998). Solving infeasibility problems in computerized test assembly. Applied Psychological Measurement, 22. [This issue]

Timminga, E. & Adema, J.J. (1995). Test construction from item banks (pp. 111-127). In G. H. Fischer & I.W. Molenaar (Eds.), The Rasch model: Foundations, recent developments, and applications. New York: Springer-Verlag.

Timminga, E. & Adema, J.J. (1996). An interactive approach to modifying infeasible 0-1 linear programming models for test construction. In G. Engelhard, Jr. & M. Wilson (Ed.), Objective measurement: Theory into practice (Vol. 3, pp. 419-436). Norwood, New Jersey: Ablex Publishing Company.

Timminga, E. & van der Linden, W.J., & Schweizer, D.A. (1996). ConTEST 2.0: A decision support system for item banking and optimal test assembly [Computer software and manual]. Groningen, The Netherlands: iec ProGAMMA.

Timminga, E., van der Linden, W.J., & Schweizer, D.A. (1997). ConTEST 2.0 Modules: A decision support system for item banking and optimal test assembly [Computer software and manual]. Groningen, The Netherlands: iec ProGAMMA.

van der Linden, W.J. (1987). Automated test construction using minimax programming. In W.J. van der Linden (Ed.), IRT-based test construction (Research Report 87-2). Enschede, The Netherlands: University of Twente, Department of Educational Measurement and Data Analysis.

van der Linden, W.J. (Ed.) (1987). IRT-based test construction (Research Report 87-2). Enschede, The Netherlands: University of Twente, Department of Educational Measurement and Data Analysis.

van der Linden, W.J. (1987). Models for use in computerized test systems. In J. Moonen & T. Plomp (Eds.), Developments in educational software and courseware (pp. 299-307). Oxford: Pergamon Press.

van der Linden, W.J. (1989). Optimaliseringsmodellen voor klassieke toetsconstructie uit een gecalibreerde itembank [Optimization models for classical test construction from a calibrated item bank]. In W.J. van der Linden & L.J.Th. van der Kamp (Eds.), Meetmethoden en data-analyse (pp. 33-42). Amsterdam, The Netherlands: Swets & Zeitlinger.

van der Linden, W.J. (1991). Toetsconstructie als een voorbeeldig ontwerpprobleem [Test construction as an exemplary design problem]. In S. Dijkstra, H.P.M. Krammer & J.M. Pieters (Eds.), De onderwijskundig ontwerper (pp. 61-70). Amsterdam, The Netherlands: Swets & Zeitlinger.

van der Linden, W.J. (1992). Selecting passage-based items for achievement tests [Internal report]. Iowa City, IA: American College Testing.

van der Linden, W.J. (1994). Optimum design in item response theory: Applications to test assembly and item calibration. In G.H. Fischer & D. Laming (Eds.), Contributions to mathematical psychology, psychometrics, and methodology (pp. 308-318). New York: Springer-Verlag.

van der Linden W.J. (1994). Computerized educational measurement. In T. Husen and T.N. Postlethwaite (Eds.). International encyclopedia of education (2nd ed., pp. 992-998). Oxford: Pergamon Press.

van der Linden, W.J. (1995). Advances in measurement: Computer applications. In T. Oakland & R.K. Hambleton (Eds.), International perspectives on academic assessment (pp. 105-124). Boston: Kluwer-Nijhof Publishing.

van der Linden, W.J. (1996). Assembling test for the measurement of multiple traits. Applied Psychological Measurement, 20, 373-388.

van der Linden, W.J. (1997). Assembling test forms for use in large-scale assessments. Proceedings of the National Assessment Governing Board Achievement Levels Workshop. Washington, DC: National Assessment Governing Board.

van der Linden, W.J. (1998). Optimal assembly of educational and psychological tests, with a bibliography. Applied Psychological Measurement, 22. [This issue]

van der Linden, W.J. (submitted). Optimal assembling of tests with item sets.

van der Linden, W.J., & Adema, J.J. (1998). Simultaneous assembly of multiple test forms. Journal of Educational Measurement. (In press)

van der Linden, W.J., & Boekkooi-Timminga, E. (1988). A zero-one programming approach to Gulliksen's matched random subsets method. Applied Psychological Measurement, 12, 201-209.

van der Linden, W.J., & Boekkooi-Timminga, E. (1989). A maximin model for test design with practical constraints. Psychometrika, 54, 237-247.

van der Linden, W.J., & Carlson, J.E. (1997). Optimizing incomplete block designs for large-scale educational assessments (unpublished manuscript). Enschede, The Netherlands: University of Twente, Department of Educational Measurement and Data Analysis.

van der Linden, W.J. & Luecht, R.M. (1996). An optimization model for test assembly to match observed-score distributions. In G. Engelhard, Jr. & M. Wilson (Ed.), Objective measurement: Theory into practice (Vol.3, pp. 405-418). Norwood, New Jersey: Ablex Publishing Company.

van der Linden, W.J., & Luecht, R.M. (in press). Observed-score equating as a test assembly problem. Psychometrika.

van der Linden, W.J., Scrams, D.J., & Schnipke, D.L. (submitted). Using response-time constraints to control for differential speededness in computerized adaptive testing. Applied Psychological Measurement.

van der Linden, W.J., & Reese, L.M. (1998). A model for optimal constrained adaptive testing. Applied Psychological Measurement, 22. [This issue]

van der Linden, W.J., Veldkamp, B.P., Reese, L.M. (in preparation). An integer programming approach to item pool design.

van der Linden, W.J., & Zwarts, M.A. (1989). Some procedures for computerized ability testing. International Journal of Educational Research, 13, 175-187.

Veldkamp, B.P. (submitted). Multidimensional test construction using Lagrangian relaxation techniques.

Verschoor, A. (1991). OTD: Optimal test design [Computer software and manual]. Arnhem, The Netherlands: Cito.

Verschoor, A. (1998). Test construction using genetic algorithms (unpublished manuscript). Arnhem, The Netherlands: Cito.

Votaw, D.F. (1952). Methods of solving some personnel classification problems. Psychometrika, 17, 255-266.

Wang, C.-S., & Ackerman, T.A. (1998). Two item selection algorithms for creating parallel test forms (unpublished manuscript). Urbana/Champaign, IL: University of Illinois, Department of Educational Psychology.

Wightman, L.F. (1998). Practical issues in computerized test assembly. Applied Psychological Measurement, 22. [This issue]

Yen, W.M. (1983). Use of the three-parameter model in the development of standardized achievement tests. In R.K. Hambleton (Ed.), Applications of item response theory. Vancouver: Educational Research Institute of British Columbia.

**Titles of Recent Research Reports from the Department of Educational Measurement and Data Analysis. University of Twente, Enschede, The Netherlands.**

RR-98-05 W.J. van der Linden, *Optimal Assembly of Educational and Psychological Tests, with a Bibliography*

RR-98-04 C.A.W. Glas, *Modification Indices for the 2-PL and the Nominal Response Model*

RR-98-03 C.A.W. Glas, *Quality Control of On-line Calibration in Computerized Assessment*

RR-98-02 R.R. Meijer & E.M.L.A. van Krimpen-Stoop, *Simulating the Null Distribution of Person-Fit Statistics for Conventional and Adaptive Tests*

RR-98-01 C.A.W. Glas, R.R. Meijer, E.M.L.A. van Krimpen-Stoop, *Statistical Tests for Person Misfit in Computerized Adaptive Testing*

RR-97-07 H.J. Vos, *A Minimax Sequential Procedure in the Context of Computerized Adaptive Mastery Testing*

RR-97-06 H.J. Vos, *Applications of Bayesian Decision Theory to Sequential Mastery Testing*

RR-97-05 W.J. van der Linden & Richard M. Luecht, *Observed-Score Equating as a Test Assembly Problem*

RR-97-04 W.J. van der Linden & J.J. Adema, *Simultaneous Assembly of Multiple Test Forms*

RR-97-03 W.J. van der Linden, *Multidimensional Adaptive Testing with a Minimum Error-Variance Criterion*

RR-97-02 W.J. van der Linden, *A Procedure for Empirical Initialization of Adaptive Testing Algorithms*

RR-97-01 W.J. van der Linden & Lynda M. Reese, *A Model for Optimal Constrained Adaptive Testing*

RR-96-04 C.A.W. Glas & A.A. Béguin, *Appropriateness of IRT Observed Score Equating*

RR-96-03 C.A.W. Glas, *Testing the Generalized Partial Credit Model*

RR-96-02 C.A.W. Glas, *Detection of Differential Item Functioning using Lagrange Multiplier Tests*

RR-96-01 W.J. van der Linden, *Bayesian Item Selection Criteria for Adaptive Testing*

RR-95-03 W.J. van der Linden, *Assembling Tests for the Measurement of Multiple Abilities*

RR-95-02 W.J. van der Linden, *Stochastic Order in Dichotomous Item Response Models for Fixed Tests, Adaptive Tests, or Multiple Abilities*

RR-95-01 W.J. van der Linden, *Some decision theory for course placement*

RR-94-17 H.J. Vos, *A compensatory model for simultaneously setting cutting scores for selection-placement-mastery decisions*

RR-94-16 H.J. Vos, *Applications of Bayesian decision theory to intelligent tutoring systems*

RR-94-15 H.J. Vos, *An intelligent tutoring system for classifying students into Instructional treatments with mastery scores*

RR-94-13 W.J.J. Veerkamp & M.P.F. Berger, *A simple and fast item selection procedure for adaptive testing*

RR-94-12 R.R. Meijer, *Nonparametric and group-based person-fit statistics: A validity study and an empirical example*

RR-94-10 W.J. van der Linden & M.A. Zwarts, *Robustness of judgments in evaluation research*

RR-94-9 L.M.W. Akkermans, *Monte Carlo estimation of the conditional Rasch model*

RR-94-8 R.R. Meijer & K. Sijtsma, *Detection of aberrant item score patterns: A review of recent developments*

RR-94-7 W.J. van der Linden & R.M. Luecht, *An optimization model for test assembly to match observed-score distributions*

RR-94-6 W.J.J. Veerkamp & M.P.F. Berger, *Some new item selection criteria for adaptive testing*

RR-94-5 R.R. Meijer, K. Sijtsma & I.W. Molenaar, *Reliability estimation for single dichotomous items*

RR-94-4 M.P.F. Berger & W.J.J. Veerkamp, *A review of selection methods for optimal design*

RR-94-3 W.J. van der Linden, *A conceptual analysis of standard setting in large-scale assessments*

RR-94-2 W.J. van der Linden & H.J. Vos, *A compensatory approach to optimal selection with mastery scores*

RR-94-1 R.R. Meijer, *The influence of the presence of deviant item score patterns on the power of a person-fit statistic*

Research Reports can be obtained at costs, Faculty of Educational Science and Technology, University of Twente, Mr. J.M.J. Nelissen, P.O. Box 217, 7500 AE Enschede, The Netherlands.

*faculty of*

# EDUCATIONAL SCIENCE AND TECHNOLOGY

A publication by
The Faculty of Educational Science and Technology of the University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands